The Visual Segmentation of Scene Information and Applications in Predictive Haptics

Bharat Srirangam

Faculty Member 1: Charles C. Kemp

Faculty Member 2: Sonia Chernova

Table Of Contents

Abstract

We as humans take our ability to digest a scene and extract its context knowledge to be for granted. There are several senses involved including but not limited to sight, hearing, and touch. This also includes our ability to combine information from the different senses to enrich our understanding. In the healthcare robotics space, a lot of success has been met with emulating or attempting to emulate these abilities of humans in everyday processes. To be specific, one process, context inference, is a very subconscious but powerful process. When someone picks up a glass cup, they can feel that the cup is made of glass where their hand meets the cup but are still able to infer that the rest of the cup is also glass. This is a powerful example of how humans take local information and use it to derive global context. In this paper, we attempt to emulate this process through a PR2 robot by developing a way to segment a scene for the various objects in the scene. The PR2 can then estimate the material of the different objects using a pre-trained neural network that takes in spectroscopy measurements and pictures of a small patch of each object. This would thereby allow the PR2 to emulate the same ability to abstract its local information to a global context as the material of each object would be determined by a small measurement. We set up a table with an arrangement of objects and test different approaches to segmenting this scene to provide points of interest and measurement to the PR2. Some objects that were used include pots, glasses, mugs, sweaters, and scissors. After testing 3 different approaches to segmentation, a 3D analysis based one was able to sufficiently segment the scenes and provided the PR2 with enough information to make proper measurements and make reasonable estimates. Finally, we demonstrate how a PR2 robot can do all of this and leverage this system to estimate the materials of everyday objects so that it can infer interactions with these objects. From this work, we find that we are one step closer to providing robots with the same advantage that we have to mix partial contextual understandings to make better globally informed decisions.

Introduction

Object detection through innovations in neural network design and computation power are specific parts of the constantly evolving fields of machine learning and deep learning. For example, the development of Convolutional Neural Networks (CNNs)[1] – specialized for ordered data recognition – and the consequential development of Long Short Term Memory Networks (LSTMs) [2], generally used in the recognition of time series data – are both examples of such developments. Other models that depend on manual features or different algorithms have also been developed such as color segmentation[10][11] which segments an image, and objects in the image, by their color. Specifically in robotics, there are also methods for 3D scene analysis[12] which allow for spatial data to determine what parts of an image are distinct objects versus just the background. While these various models and methods are powerful tools and have their own strengths in different tasks like image recognition and classification or scene segmentation, none of them have been as well explored or incorporated in the field of haptic data and haptic recognition.

Large advancements and procedures associated with computer vision have resulted in a near perfect ability to generalize recognition performance in a way that can be applied to products and services everywhere. An example would be how recent laptops and other handheld devices have been able to commit to using facial recognition as a form of security for any user that purchases a device - that is to say with only a small amount of data from the user. These sequential advancements for image recognition leave a solid roadmap for the task of haptic recognition, which is fundamentally similar to the task of image recognition. Rather than just focusing on the recognition capabilities that humans have been able to gain with sight, we can focus on harnessing the abilities of the other senses such as touch and hearing. By combining the information derived from both areas, we can give actors in a robotics or simulated environment a more complete set of context information that humans normally have.

One of the best applications of our ability to see is that we can abstract what we remember from the past and apply it to recognizing scenes, images and objects we have never seen before. This application of prior knowledge or previous history is well searched in the field of computer vision – ie. recognizing whether an object is a table or not - but not as closely studied in the space of haptic data and haptic science. Some recent research has begun to search this space of how to abstract local information to derive global information. For example, Erikson et al. focused on understanding the global haptic layout of a scene by measuring specific force and haptic information of local areas along a specific part of the scene. They then proceeded to use computer vision to infer the haptic information of other parts of the scene that had similar features and colors[3]. This allowed the robot from the study to effectively generalize its knowledge to understanding a whole scene layout from only a local sample of information. It is important to note that there are several approaches to gathering information about the scene which can have various levels of success. Once a object segmentation process for scenes are chosen, we can develop an intertwined computer vision and haptic recognition method to

generalize scene context information - information that has not been previously leveraged as well in the applied robotics space.

Specifically, we propose the creation of an environment that might simulate the types of scenes that haptic and visual contextual information could be combined to provide behavioral insight - for example, a kitchen table with different objects that could be used or interacted with. The various objects - of different colors and materials - would make up a scene of objects on a table. Then, using the most effective segmentation process, the segmented objects in the scene combined with a 3D data map of the same scene, allows us to find a surface normal to every object of our image. A robot's arm could then approach each object along the surface normal with a small spectroscopy camera attached to its end to take pictures and measurements [7]. Using an existing mapping of spectroscopy measurement to material label[7], the final step in the process would be to infer the material label of the objects in the image based on the object segmentations created. Like in computer vision, this allows us to take smaller local measurements of the scene to generalize the known information to the whole scene.

This ability to generalize can help in several areas of applied robotics – specifically, in the health care robotics space. The ability to recognize objects that have not been seen before is fundamental to this generalization of context information as it provides the foundation to infer the physical properties of objects in a scene. This context knowledge then allows robots a wider range of options when interacting with these objects, the same range of options that humans can choose from. For example, a human can pick up a never before seen bowl and infer that it is a glass bowl from prior knowledge and experience. This type information is important if, for example, we want to know what objects are microwave safe, etc. The comparison of the various methods for scene segmentation provides a nice research space to find the best segmentations to be used in haptic and visual scene recognition.

Literature Review

There are two strains of work that are explored in the presented research: the actual segmentation of a scene and the actual application of combining visual and haptic data to provide an estimated haptic reading for various parts of a scene. In terms of the scene segmentations, several approaches have been adapted such as deep learning approaches, manually made features, and even 3D analysis. The use and viability of these different approaches are differentiated both on actual instruments or algorithms used as well as the applicability to various tasks.

Over the past several years, there has been a lot of work done in the space of object detection. However, for the sake of the environment used in this work, we shall be reviewing

only work done on object detection with the body of the object used as a marker rather than a simple box around the image - consider the images below.

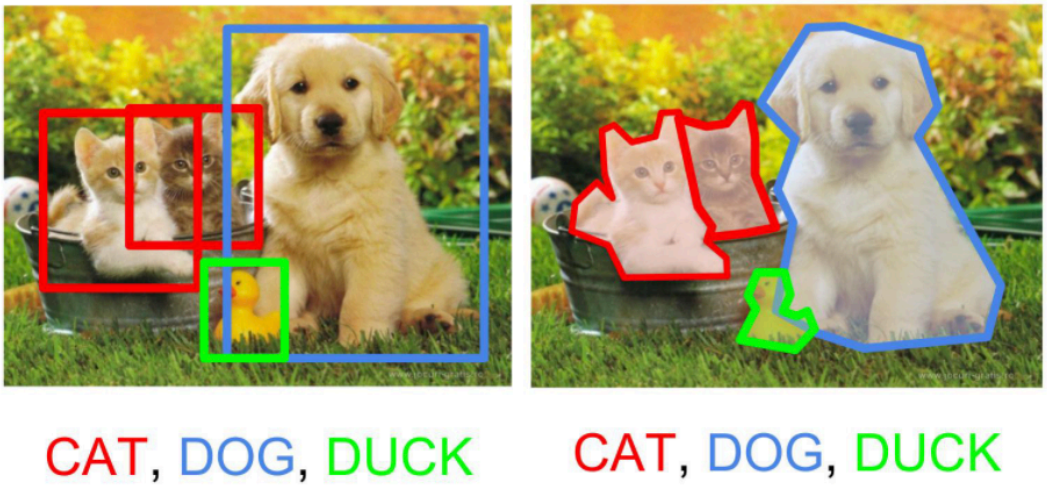

Figure 1: Localization vs Object Detection

The image on the left is an example of how localization with a box of the objects in the image are used for object detection while on the right, the full body of the objects are used to mark the objects themselves. Research into the models that produce the information like that on the right will be reviewed. The three main areas of related work we wish to focus on are a Deep Learning review, Color Segmentation, and 3D Analysis.

Color Segmentation:

Color segmentation is a well searched approach that was built upon the monochromatic images of earlier times. Color images provide a greater source of information than Gray images and don't have any added work that needs to be done. While color images are a bit more computationally intensive than gray images, the difference is negligible with today's technology. There have been feature based approaches on segmenting RGB images which are then projected to 3D space[13][14]. There have also been various clustering techniques for segmenting scenes in images, based on the colors of the different parts of the scenes[15].

In this experiment, the watershed algorithm is the main part of the color based segmentation approach. This is the same watershed algorithm invented by Meyer for signal processing[16] and which has also been adapted for color image segmentation[17]. The main takeaway from their work is the applications of their algorithm - ours is intended to find the entire body of objects while theirs intends to find regions of connected similar colors. One issue with this approach is that for objects with several different colors, these objects will be split into several different regions. This will be discussed in more depth in the discussion.

3D Analysis

There have been complex features developed based on descriptors from 3D point clouds themselves rather than through color segmentation [23]. These features are not without issue though. The time needed in creating these descriptors and complexity of the analysis can increase to non realistic runtimes which are not suitable for our problem space. Other work that is related to our experiment include the segmentation of 3D point clouds for household environments[24] and complex object detection[25]. While both works use complex and effective methods for their tasks, the main difference from our work is the task itself and use of information. While these tasks are targeted towards finding specific parts of scenes or the entire analysis of a scene, our work is focused on finding objects - that are convex in shape - that can be manipulated or measured for the accumulation of context knowledge. For example, we would want the information provided by finding the objects on a kitchen table but not the table itself. Another separate but important difference is the tools used in our work versus those of the other papers.

The work that most resembles our 3D analysis approach is the work done by Lopez where they detect objects using the color and depth features with a Kinect sensor [26]. The key difference between their work and approach from our experiment is the fact that their task is different from our task which leads to other small but important differences. They are interested in robotic navigation which means they intend to focus on specific objects in a specific direction or within a specific vicinity while our work focuses on segmenting a scene for all objects in the area. Their approach also allows them to focus on specific objects while our work tries to collect information on all objects in a region of the image - such as all the objects on top of the kitchen table - without reference to the kitchen table itself. The approach outlined in this report draws on inspiration from these previous works to determine the best way to find the different objects.

Deep Learning Review

While color segmentation and even 3D analysis have come a long way as manual features, some features have been learned by employing networks such as CNNs. For example, in Gupta et al. and others, CNN's are used to learn RGB-D features to help with object detection[18][19][20]. Oftentimes however, learning these features can sometimes become dataset specific if the background of the dataset is used as a different class from the objects. Some other work has begun to include networks that have been trained in an unsupervised fashion such that they can become dataset neutral and act as plug in replacements for methods. Regular Artificial Neural Networks have also been used in object detection as a way to create unique color-pixel based features. Their accuracy, ability to generalize and speed have led ANNs to have their own success [21]. Surveys over deep learning in object detection have also connected a lot of work that has been done and made two general categories for some of the different types of object detection [22]. One is where a specific object is being searched for while the second is for more generic or unseen objects such as a car or person. While these are very fundamental problems in computer vision, the goal of this work is to provide context knowledge

for any scene which means any arrangement of any object should be able to be segmented. This key aspect will play into decisions that are made later in this report.

The deep learning approach we attempted to use in this report is based off of the Mask-RNN model based off of the work done by He[27] and the implementation done by Abdulla [28]. Note that this work was used in hopes of trying to attain a viable scene segmentation that could be used to accumulate a general context knowledge of the haptic readings of the scene - or just the important parts of the scene such as the objects.

Haptic Recognition

The field of haptic data recognition, which originally has not received as much attention as the visual data recognition field, has had recent advancements – in particular with the conjunction of deep learning as opposed to artificially created filters. The first focus on material recognition or haptic recognition in general began with Bell et al's introduction to the Materials in Context Database (MINC) for material classification of everyday images [4]. However, this class of problems and suggested solutions does not address every real life situation – for example, inferring whether an object is metal or microwave safe. Bhattacharjee et al. introduced the use of supervised learning to material recognition using haptic data such as active temperature sensing over time [5]. Using different frameworks such as SVM's with different interaction times, their robot achieved accuracies of 84% and 98%. However, some limitations with their work include not being able to deal with uneven surfaces and failing to get a proper distribution of interactions that a robot would actually have in real life.

The closest and most relevant work is the work done by Erickson et al. for which this work is related to and continuing off of [6]. In the research, the augmentation of object and scene recognition with measured haptic information is used as a way to infer the same measurements of other unexplored parts of the environment - essentially developing context knowledge. The experiment that was conducted had the robot wear a haptic sleeve that allowed it to record different modes of haptic data, such as force and temperature and take measurements along a specific trajectory in part of the visible scene. This information was treated as a sort of label which was applied to unexplored patches of the image. This allowed the robot to match that specific information to the exact scene that the robot was currently processing. This experiment is the closest setup to our experiment in published research but what makes our approach unique is the combination of different haptic measurements and a survey over different scene segmentations. Besides the actual system of segmentation and mapping, the other important change in our experiment is the use of a spectroscopy camera instead of a haptic sleeve as a tool for measuring haptic data in the image[7]. In Erikson et al, he shows how using a spectrometer to collect spectroscopy data and training a neural network with this data and the appropriate labels gives a function that is actually quite successful at predicting what materials that you are looking at in different objects[7]. This successful use of haptic measurement tools provide a great opportunity and motivation to combine this form of measurement with modern computer vision techniques to provide a connection between the haptic data of an object and what it looks like.

The distinct and important difference between the work in Erickson et al.[6],[7] and our work is the combination of scene segmentation and spectroscopy measurement as a way of collecting haptic data and inferring general scene information. In addition to this, our work focuses on also surveying some of the different ways to segment the scene for the objects. The use of spectroscopy in our process of inferring the material classification of different objects is likely to provide more accurate results overall. Spectroscopy is able to give a much better representation of the materials since the actual data retrieved is actually based directly on the materials rather than the shape or temperature of the objects[9]. With the use of pretrained models that can take in the spectroscopy measurement, small picture of the material, and a proper scene segmentation, we are able to develop the most accurate collection of context information over the various objects in a scene which has hardly been explored to date.

Methodology

Since this work is based on various approaches and their applications, my research is in the unique position of having multiple trials of different tests instead of repeated trials of the same tests to confirm and validate results. It is assumed that the setup of different algorithms and programs are deterministic (unless the research itself is in the space of randomness). This specific type of research is the based on robotic perception of the outside world, specifically object and color segmentation of images as well as the spectroscopy measurements involved with those objects. After designing and creating an implementation for each of the segmentation approaches, the ability for each approach to provide a scene with the key objects highlighted or segmented from the rest of the image can be compared. Once the best approach is chosen, the rest of the experiment is actually based on testing this algorithm on a PR2 robot by having it go through a study of recognizing different objects. These objects will have different colors, shapes, and sizes but the goal is to have the PR2 determine the material of the different objects through a single measurement for a small part of the object. The various implementations and such are described below. The experiment description with the PR2 follows.

Color Segmentation Algorithm (Approach 1):

This algorithm is based on the process of taking an image and finding the explicit regions of same or “similar” color and treating them as the same region or part of the image. This is like object detection in a sense where different parts, or pixels, of an image that look similar and are located next to each other are considered part of the same object – thereby forming regions. The code that accomplishes this is written in Python 3 and uses the python packages of OpenCV, NumPy, and SciKit Learn. The code below walks through the major steps in the algorithm, minor details are commented out in the quotations.

```
def flood_fill_approach(self, img):
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (7, 7))
    gradient = cv2.morphologyEx(blur, cv2.MORPH_GRADIENT, kernel)
    """""""
    binary = cv2.inRange(gradient, lowerb, upperb)
    """""""
    foreground = cv2.morphologyEx(foreground, cv2.MORPH_CLOSE, kernel)
    """""""
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (17, 17))
    background = cv2.dilate(foreground, kernel, iterations=3)
    unknown = cv2.subtract(background, foreground)
    """""""
    markers = cv2.connectedComponents(foreground)[1]
    """""""
    markers = cv2.watershed(img, markers)
    """""""
    marker_img = cv2.merge([hue_markers, blank_channel, blank_channel])
    marker_img = cv2.cvtColor(marker_img, cv2.COLOR_HSV2BGR)
    """""""
    set_of_labels = (stats[stats[:,-2] > threshold_for_coallessing])[:, -1]
    markers, all_cluster_centers, _ = self.adjustable_groupingv2(markers, ordering, cluster_centers, other_centers, stats)
    final_clustercenters = self.find_final_clustercenters(set_of_labels, all_cluster_centers, stats)
```

Figure 2: Color Segmentation Pseudocode

The first step is to create the filters to run over the image to get rid of noise in the data and make the algorithm more deterministic. The next step then creates a binary image which allows us to create a background and foreground image. With the difference of the background and foreground images, we can find the markers of connected components in the overall image and use the *watershed* openCV algorithm to calculate some preliminary color regions. We can then proceed to color the original image with these color regions and combine the smaller ones as necessary into larger regions. The actual algorithm was run and tested on a Mac Book Pro with an i5 processor and 8 GB of ram. It was tested on images such as Figure 6 and Figure 8 which provide some basic house hold objects that a robot might have to interact with. Note that these testing images were taken by an Iphone 6s.

3D Analysis Segmentation (Approach 2):

This approach is based on taking a paired color image and 3D image to find all parts of the color image that are above the table in the scene. The table is manually determined in the image so that the algorithm is able to determine relative "above" and "below" pixels in the image. Once all the pixels above the table have been collected, a single link agglomerative clustering algorithm with a max cluster distance of 2 is used to create distinct clusters of "above table" pixels. Each cluster is considered an object and returned as the objects in the image/scene. The code that accomplishes this is written in Python 3 and uses the python packages of Open3D, OpenCV, NumPy, and SciKit Learn. The pseudocode below walks through the major steps in the algorithm.

```
def threeD_analysis:
    points3D = data.reshape().crop()
    points3D.drop_bad_points() #all missing 3D points are given base "0" values
    points3D.fillin_image() #Smoothens the 2D image to fill in any missing values

    relative_position_matrix = analyze(points3D, table_index, fudge_factor) #gives a 2D matrix of T/F values for whether
    the pixel is above or below the given table_index with an added fudge_factor.
    above_table_indicies = points3D.get_above_table(relative_position_matrix) #gets all pixels that are measured to be
    above the table.

    clusters = cluster(above_table_indicies) #clusters the pixels above the table using Single Link clustering with a
    maximum distance of 2 between clusters.
    points3D, kinect_image = color_image(clusters) #update a colored version of the original image to segment all objects
    above the table.
```

Figure 3: 3D Analysis Pseudocode

The first step is to properly clean up the 3D image by reshaping and cropping it properly and filling in missing values from errors in the camera. After this, provided a hard coded location and dimensions of the table, we can calculate a second 2D image that has a true or false value for each pixel which tells us whether it is above the table or below the table. Note that a fudge factor is added to the height of the table so that noisy measurements by the Kinect do not hinder performance. The pixels are then clustered into "objects" and then the image is colored for each "object". The actual algorithm was tested on a Mac Book Pro with an i5 processor and 8 GB of ram. The actual experiment code executions and all calculations were done on the PR2. Some example scenes include those in Figure 10 and 12. Note that these images were taken by an Xbox One Kinect.

Deep Learning Segmentation (Approach 3):

This approach is based on using previously trained "neutral" neural networks that can take in an image and find the various objects of potential interest in the image by highlighting them. The model is loaded into the program and then the color image is cleaned and reshaped as necessary to be passed through the model. The model will then output a mask and this mask can tell us which pixels in the image need to be colored - as in which parts of the image are considered "object" or not. The code that accomplishes this is written in Python 3 and uses the python packages of Keras, OpenCV, NumPy, and SciKit Learn. The pseudo code below walks through the major steps in the algorithm.

```
def mask_rnn():
    image = data.reshape().crop() #Clean and reshape the image
    model = load_model() #Load a pre-trained model to create a mask from the provided image

    mask = model(image) #Get a mask for the image
    image.color_objects(mask) #Color the objects in the image via the mask created by the model
    return mask #Use the mask to find object centers
```

Figure 4: Deep Learning Pseudocode

The first step is to clean up the image through reshaping and cropping. The pre-trained model is then loaded into the program. The image is pushed through to create the mask which is then used to color the image appropriately. The mask can then be used to return the information about where the various objects are and create a representation of a scene segmentation. The actual algorithm was run and tested on a Mac Book Pro with an i5 processor and 8 GB of ram. It was tested on images such as Figure 14 which provides some basic house hold objects that a robot might have to interact with. Note that these testing images were taken by an Iphone 6s.

PR2 Robotic Study with Spectroscopy:

Once an algorithm has been chosen to segment the images that a robot might see, a study can be conducted with a robot, such as the PR2, that explicitly walks through the process of estimating the material of different objects. We can set up the apparatus of the experiment to be fairly simple. First we take a simple table, in our apparatus we chose a single colored white table,

and take a collection of household objects to have the robot interact with - overall resembling what might be on a kitchen table. Dishes, cups, plates, and blocks are examples of different objects that can be incorporated into the study. We then place this table in front of the PR2 robot and have the Xbox One Kinect sensor attached to the robot take a picture and 3D depth picture of the table with the objects on it. This image is then pipelined through the algorithm that was chosen and returns different areas on objects. At this point, the PR2 takes each area and finds the surface normal to that object at its center and uses its end effector to approach the object with a spectroscopy camera, a Lumini sensor and a SCiO sensor combined, to make specific measurements. Once all these measurements have been taken, they are run through a pre trained neural network from a previous study[4], and an estimated material label is given to that specific region. Parts of the objects that were not measured can be given the same label as the respective measured sections. This will act as the inference of context information on the full bodies of different objects within the image. The actual objects themselves are completely randomly chosen from a set of common house hold items – especially ones located in kitchens. As our lab focuses on health care robotics, these are the areas that most of our robots will be located in which means that these objects are also the most common objects that our robots would have to interact and deal with.

```
def measure_objects():
    #Given the clusters in the image
    normals = o3d.geometry.estimate_normals(points3D) #Calculates all normals to all pixels in the 3D image
    object_normals = get_normals(clusters, normals) #Get the normals to each clustered object.
    approach_calcs = get_rpys(object_normals, clusters) #Calculate all approach measurements for each object.

    take_measurements(clusters, object_normals, approach_calcs) #Navigate the PR2's end effectors to take measurements of
    each object
```

Figure 5: PR2 Measurement Pseudocode

Above is the pseudocode involved in the PR2 making measurements during the experiment.

Success Metrics:

There are two fundamental parts to the experiment above. The first is the ability to segment the scene for all the objects present in a useful way. Success for this would be measured by whether all the objects have their own color. While pixels that should be colored, missing colored pixels, and pixels that should not be colored, incorrectly colored pixels, are both incorrect or signs of inaccuracy - it is more beneficial to have missing pixels over incorrect pixels as a false positive is more harmful than a false negative for calculating locations and trajectories. That being said, fringe pixels such as those along the edges of the objects are not as impactful as the ones at perhaps the edge of an image. So as long as each object has its own color, then we know that they were all recognized. As for the actual apparatus in the second part of the experiment, the PR2's ability to properly take spectral measurements of the different objects and use the pre-trained neural network to properly categorize those objects would be the success metric. It is important to note that the accuracy of the pre-trained neural network is not part of the success metric and that just the process of estimating a material for each object

successfully is all that matters. These metrics can be changed as necessary if a different definition of success is desired.

Rationale:

While several design decisions were made based off monetary constraints or industry standards, there were some major decisions and choices that had unique rationales. For example, the table that was used in the experiment was chosen to resemble a kitchen table to better emulate the real life applications of this research. In our specific experiment, the table was white out of convenience. The real requirement for the table or surface in general is that it could be found in a common kitchen or home. The reason that an Xbox One Kinect was used was because the PR2 Robot had a documentation guide on incorporating the Kinect onto the PR2's "onboard" cameras. This also allowed us to make measurements of both 2D images and 3D depth images over the same scene without needing any additional calibration. The decision for which approach to take for object segmentation is a direct result of the successes from the first part of the experiment where different approaches are compared. Obviously there are a large number of different approaches that could have been chosen, but the reason these specific approaches were chosen were based on personal experience with the different algorithms as well as most promise based on related works. It is difficult to expect robots to necessarily do better than humans in specific tasks such as this one when the whole system is designed on the way humans tackle these tasks. Several implementation decisions were made based on the availability of open source libraries (OpenCV, SciKit Learn, etc) and personal experience with such code bases. Finally for testing, more house hold items were chosen at random but in the theme of objects that might be found around a kitchen to test the realistic bounds of the system and robot.

Results

The results of my two part experiment are a collection of different images. These images are relevant for the assessment of whether this experiment was successful or not because they provide the answer to the question of whether our different approaches worked or not. There are two sets of results to report - the output of the 3 various approaches that were explored to provide proper scene segmentations and the actual experiment with the PR2 to use the scene segmentation to provide measurements of the scene to create a generalized context knowledge. To see potential success in all parts of the experiment we can see the results for each approach and determine if the objects of the image were appropriately captured and segmented in the image. If an appropriate center to the object could be viably determined then we know that this information could be used by the PR2 to complete the second half of the experiment. The success of the second half of the experiment would be based on the best approach from the first half and then whether the PR2 was able to emulate the behavior of a human to measure small

patches of different objects to determine an entire context knowledge for the whole scene. Below are the results of the experiment.

Color Segmentation Algorithm (Approach 1):

For the first color segmentation approach, different scenes were tested for proper segmentation. Below are these conversions from a regular color image to their segmented versions.

Figure 6

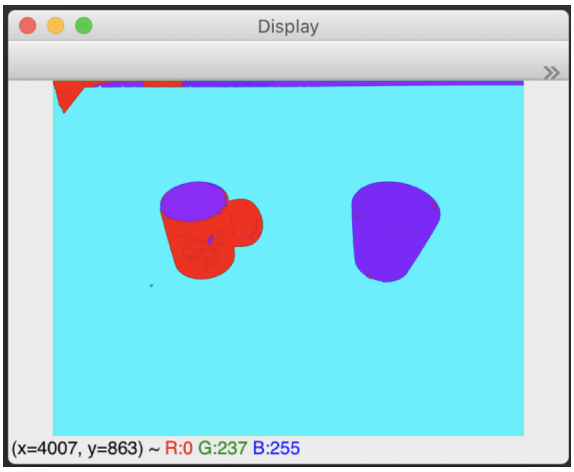

Figure 7

As we can see, in the scene above, a “kitchen table” has a small pot and cup that may be interacted with. After running the image on the left through our color based segmentation algorithm, we can see that the table was successfully distinguished from the objects themselves. As we can see, the small pot, being a single color, was distinguished as its own object while the cup was perceived as two regions - one on the inside and one on the outside because of their starkly different colors. In the background, there are more colors that are combined into “unreachable” regions as they cover areas that are not within grasp of the robot. Below is another scene that was tested.

Figure 7

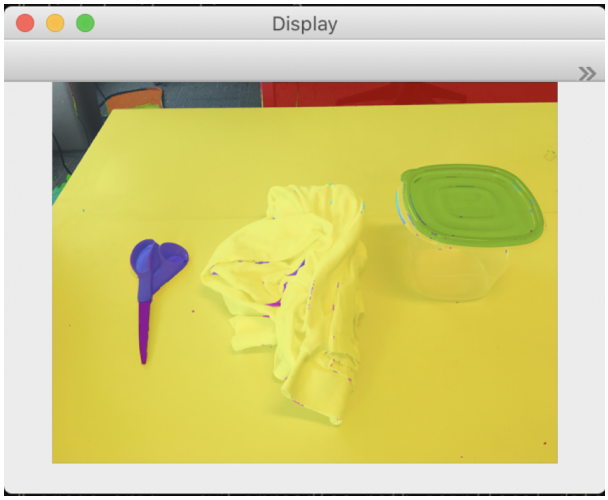

Figure 9

As we can see in this pair of images, the scissors, both the handle and metal blades, were recognized as separate colors on the object of scissors and as a result were colored differently. The white sweater on the table is not recognized as the background table is too similar in color that the color segmentation algorithm is not able to properly distinguish it from the background. We can see that it was imprecise from the highlights of purple and pink on the sweater in the image on the right. Finally, the third object on the right is split into a container and its lid. The container is transparent and thereby it is likely that the algorithm was not able to distinguish the container from the table. The lid, however, is red and a distinct color in front of the white table and was successfully colored. The remainder of the image, the background, has miscellaneous colors but does not affect the actual segmentations of objects on the table and is therefore irrelevant. It is important to note that the various colors and amount of light affected performance - this will be explored in the discussion.

3D Analysis Segmentation (Approach 2):

For the 3D analysis approach, different scenes were also tested for proper segmentation. Below are these conversions from two regular color images to their segmented versions.

Figure 10

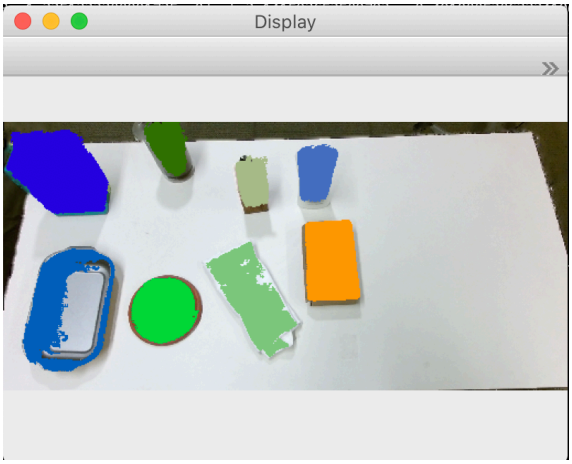

Figure 11

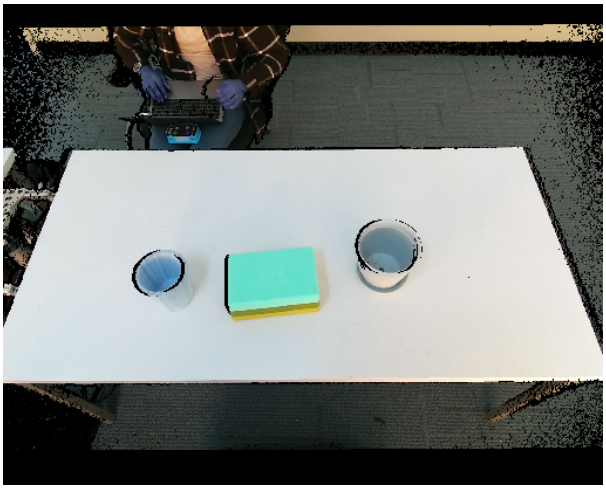

Figure 12

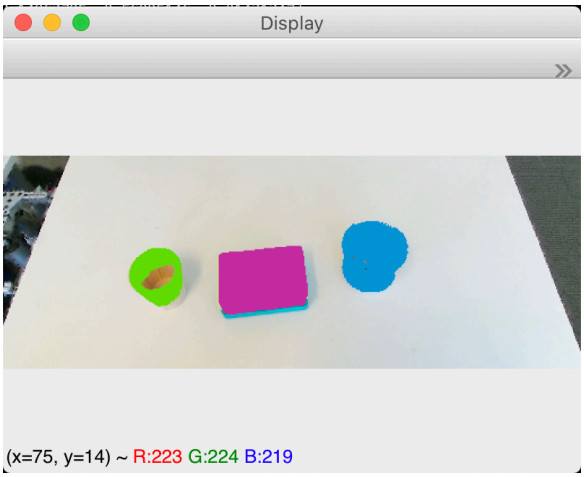

Figure 13

In the four above images, the assortment of household items are spread out on the "kitchen table". We can see that there are some black pixels throughout the images on the left side and these come from imperfections in the Kinect camera. The left side images represent the scene while the right side images represent the segmented images. We can clearly see, in both segmentation images, that only a single color is used to color over each of the objects - even objects with multiple colors. The only issue is that for some objects, such as the mettle pan or wooden pot, not all of the object is colored. As discussed in the methodology, it is much better to have false negatives than false positives in terms of determining whether a pixel is part of an object or not. With that said, one reasonable explanation for why these parts of the image are not classified as part of an image could simply be because the reading of that part of the object fell below the fudge factor. As we can see, overall, each object has its own color and has been "segmented" from the scene - presumably with the important information passed to the PR2.

Deep Learning Segmentation (Approach 3):

For the deep learning segmentation approach, a single scene was tested for proper segmentation - the results did not warrant further testing which will be shown in the discussion. Below is the conversion from a regular color image to the segmented version.

Figure 14

Figure 15

As we can see, in the scene above, a "kitchen table" has various items on it which might be found in a regular kitchen. The image on the left, when run through the mask-rnn deep learning system, produces the image on the right. As we can see, the bowl seems most captured by the network as well as part of the coffee canister on the right of the wooden bowl. However, very little of the glass cup and metallic tumbler were captured by the network. In addition, there seems to be part of the wall that was considered "an object" which counts a false positive which is very detrimental to the PR2's process. On top of that, the objects that were given a mask did not have a clean single color mask but rather were a collection of different colors suggesting that the PR2 would interpret the mask as having way more objects than actually present. There are a

lot of varying factors such as shading, lighting, and even angle. The training for the network might have also affected results but this will be analyzed in the discussion.

PR2 Robotic Study with Spectroscopy:

As well be shown later in the report in the discussion, the second approach was deemed the most effective in creating segmentations for different scenes. Now with this approach, the PR2 was able to use this information to determine how many objects were on the table and where to take measurements of a small part of each object to infer its material composition. Using each measurement and a pre-trained neural network, an estimate for the material of each object was made and designated to the whole object. Below in Figure 16, we can see a scene that was segmented and then the general context knowledge about the scene that was constructed by the PR2.

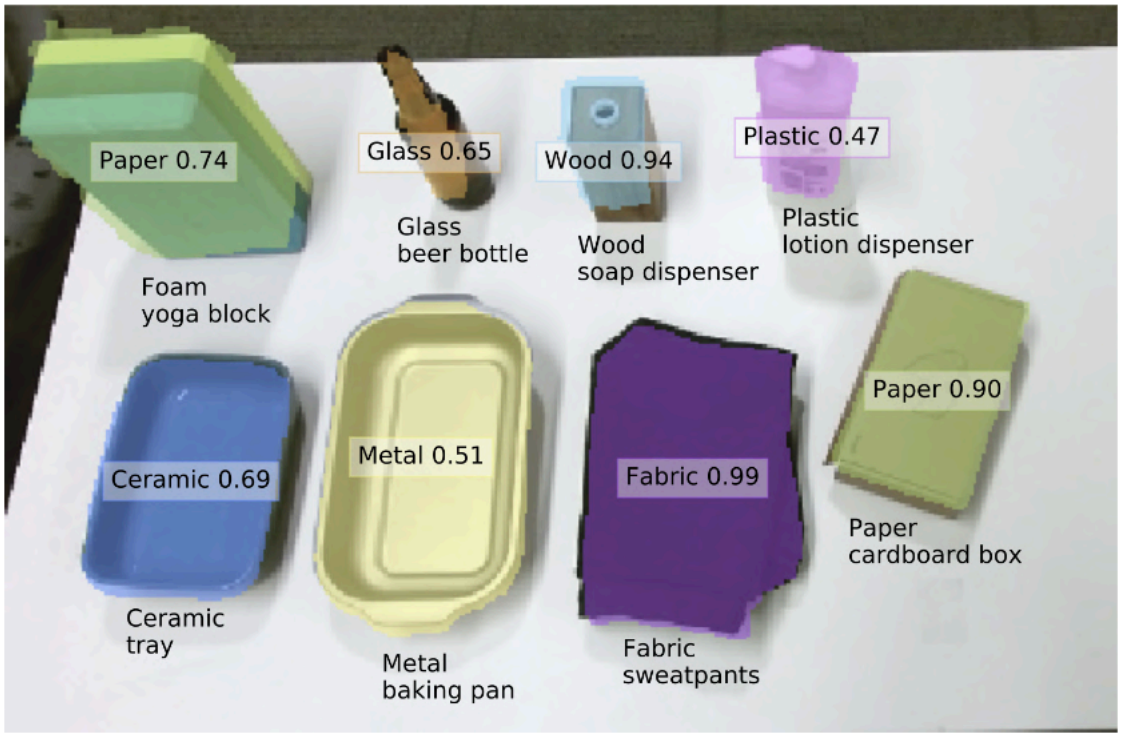

Figure 16

Discussion (Conclusion)

The results for the various parts of the experiment are described above in detail but, how do we know whether the results are actually useful? What parts of the experiment were successful and what parts were not? How do we know we don't need to do better than we already have? These are important questions and must be tackled one at a time. For the Deep Learning approach, results were fast and practical in future tasks but the actual segmentations themselves were very poor for the experiment that had been set up. The various colors over the same object would be difficult to combine into a single color for a single arbitrary object. In addition to this,

two objects were also mostly missed by the network. On top of this, there was an additional "object" detected where the wall was providing a false positive which would have been very costly in the second half of the experiment with the PR2. To reiterate, false positives are more difficult to deal with than false negatives because each positive represents a place the PR2 might try to measure in the scene which risks messing up the environment created. False negatives however are not necessarily as dangerous. For example, if an object has some pixels considered just part of the table, the other properly segmented parts of the object provide enough information for the PR2 to still measure them. Any noise on the edge of the objects would be taken care of by the PR2 choosing an average point during measurement. While there is potential to properly train the network to recognize the objects that might be interacted with, it is shown later on that other methods of segmentation achieve this and do not have the same sensitivity to light and shading that a Deep Learning approach might have.

The Color Image Segmentation approach was met with more success than the Deep Learning approach. As was shown in the results section, objects were reasonably detected and properly segmented which would provide the valuable information to the PR2. However, some issues include the multiple segmentations of a single object - if it was made of several colors. This potentially cause multiple places of measurement for the PR2 that would be unnecessary and potentially difficult. For example, the Buzz Mug in the first scene is segmented by its inside and outside simply because the inside is yellow. This would be very difficult for the PR2 to navigate on an arbitrary object. The same sensitivity problems also follow this approach in that light and shading can drastically impact performance. So while this approach had some interesting results and did not suffer from false positives like the previous approach, a different approach might still yield more success.

The 3D Analysis Segmentation approach had the most success with all the objects in both synthesized scenes being properly segmented. If we look at the results for these segmentations, we can see that all objects were segmented without any false positives. While there were certainly some false negatives with pixel level classification as "object" or not, the pixels that were properly classified provided enough information to the PR2 to still consider location and take the proper measurements in the second half of the experiment. In fact, this means that any object that is convex in shape would not suffer from some false negatives. If a object had a non-convex shape, then a point of measurement would need to be localized to the nearest "known" part of the object. The benefit to using a 3D image over the color based images that are fundamental to the other two approaches is that lighting and shading do not impact performance. On top of this, the color and shape of objects also do not affect the performance either. This essentially gives this approach near perfect ability to generalize to any object and is only limited by the granularity of the 3D image. With no training and a relatively fast computation time, this approach serves as the best way for the PR2 to segment a scene of objects.

As described in the Methodology section, the second half of the experiment was to get the PR2 to effectively use a segmentation of a scene to take measurements of objects in the scene. A small patch of each object was recorded - an image and spectroscopy measurement -

and after being classified by a pre-trained model used and designed by Erikson[7], the entire object was considered a specific material, which allowed the creation of generalized context knowledge. In the results provided, we can clearly see that the PR2 was able to use the segmentation to take measurements of each object. Even though each object is given an estimate based on the classification of a pre-trained neural network, the actual classification is irrelevant in the actual success of this part of the project. Because the PR2 was able to leverage a segmentation of the relevant scene and build up context knowledge (whether right or wrong is based on other separate factors), we can consider this a success.

It is important to remember the purpose of this research is to explore the application between haptic recognition and computer vision in health-care robotics. With that said, we have now been able to successfully emulate the process a human takes when an object is observed and material is estimated. We are able to infer the material in one part of the object and assume that the material is the same for the rest of the object, and finally are able to compile a more complete context knowledge of the scene. In this case, this is the table with all the objects.

The room for improvement for the experiment lies in very specific areas. First is in the speed of the whole system which is primarily bottlenecked in the actual segmentation of the image. This can change based on different clustering algorithms or hardware changes. Hyperparameters for the different algorithms can also be tuned such that the minimal yet sufficient amount of work can be done. We can focus on speed as that becomes necessary for future improvements. We can also improve the actual neural network that is used for the material recognition aspect in order to better simulate the behavior of humans. This is however separate from the actual process of improving the robot's performance because this is a separate problem to solve in general. The next steps of work in this project would be to focus on generating other context knowledge of the environment that could be potentially useful for the robot. This might include the sound context of the scene. Perhaps additional segmentation approaches could be explored or combined. We could also focus on getting context knowledge in dynamic scenes and non controlled scenes to determine the extent to which a robot can emulate a human's inference skills.

Some things I would have changed with this study is the various scenes used for testing the different approaches. It would have been easier to compare performance between each approach if the scenes were all the same but instead, the reality is that each approach was tested in sequential order till a viable one was found. I do not believe this compromises any integrity to the work as the available results still speak to the approaches ability to develop valid scene segmentations. I would have also explored more diverse segmentation options if given more time and resources. This might have provided a better path to gaining context knowledge than our current method. As the research progresses forward, robotics will begin to explore the more and more complex interactions of different sensory modes with each other rather than individually. Ideally, we can create a robot that can emulate a human's actions perfectly.

Citations

1. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105)
2. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.
3. Bhattacharjee, T., Shenoi, A. A., Park, D., Rehg, J. M., & Kemp, C. C. (2015, September). Combining tactile sensing and vision for rapid haptic mapping. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1200-1207). IEEE.
4. S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. Computer Vision and Pattern Recognition (CVPR), 2015.
5. T. Bhattacharjee, J. Wade, and C. C. Kemp. Material recognition from heat transfer given varying initial conditions and short-duration contact. In Robotics: Science and Systems, 2015.
6. Bhattacharjee, T., Shenoi, A. A., Park, D., Rehg, J. M., & Kemp, C. C. (2015, September). Combining tactile sensing and vision for rapid haptic mapping. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1200-1207). IEEE.
7. Erickson, Z., Luskey, N., Chernova, S., & Kemp, C. C. (2019). Classification of household materials via spectroscopy. *IEEE Robotics and Automation Letters*, *4*(2), 700-707.
8. Forsyth, D. A., & Ponce, J. (2002). *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference.
9. Ingle Jr, J. D., & Crouch, S. R. (1988). Spectrochemical analysis.
10. Lucchese, L., & Mitra, S. K. (2001). Colour image segmentation: a state-of-the-art survey. *Proceedings-Indian National Science Academy Part A*, *67*(2), 207-222.
11. Vantaram, S. R., & Saber, E. (2012). Survey of contemporary trends in color image segmentation. *Journal of Electronic Imaging*, *21*(4), 040901.
12. Nguyen, A., & Le, B. (2013, November). 3D point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)* (pp. 225-230). IEEE.
13. Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.
14. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91-110.
15. Tan, K. S., Isa, N. A. M., & Lim, W. H. (2013). Color image segmentation using adaptive unsupervised clustering approach. *Applied Soft Computing*, *13*(4), 2017-2036.
16. Meyer, F. (1994). Topographic distance and watershed lines. *Signal processing*, *38*(1), 113-125.
17. Saarinen, K. (1994, November). Color image segmentation by a watershed algorithm and region adjacency graph processing. In *Proceedings of 1st International Conference on Image Processing* (Vol. 3, pp. 1021-1025). IEEE.
18. Gupta, S., Girshick, R., Arbeláez, P., & Malik, J. (2014, September). Learning rich features from RGB-D images for object detection and segmentation. In *European conference on computer vision* (pp. 345-360). Springer, Cham.
19. Gupta, S., Arbeláez, P., Girshick, R., & Malik, J. (2015). Aligning 3D models to RGB-D images of cluttered scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4731-4740).

20. Wohlhart, P., & Lepetit, V. (2015). Learning descriptors for object recognition and 3d pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3109-3118).
21. Hassanat, A. B., Alkasassbeh, M., Al-Awadi, M., & Esra'a, A. A. (2016). Color-based object segmentation method using artificial neural network. *Simulation Modelling Practice and Theory*, *64*, 3-17.
22. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International journal of computer vision*, *128*(2), 261-318.
23. Mian, A., Bennamoun, M., & Owens, R. (2010). On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, *89*(2-3), 348-361.
24. Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., & Beetz, M. (2008). Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, *56*(11), 927-941.
25. Huang, J., & You, S. (2013, June). Detecting objects in scene point cloud: A combinational approach. In *2013 International Conference on 3D Vision-3DV 2013* (pp. 175-182). IEEE.
26. Hernandez-Lopez, J. J., Quintanilla-Olvera, A. L., López-Ramírez, J. L., Rangel-Butanda, F. J., Ibarra-Manzano, M. A., & Almanza-Ojeda, D. L. (2012). Detecting objects using color and depth segmentation with Kinect sensor. *Procedia Technology*, *3*, 196-204.
27. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).
28. Waleed Abdulla. 2017. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN
29. Erickson, Z., Xing, E., Srirangam, B., Chernova, S., & Kemp, C. C. (2020). Multimodal Material Classification for Robots using Spectroscopy and High Resolution Texture Imaging. *arXiv preprint arXiv:2004.01160*.